

## CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method, comprising:  
initializing a debugger in a first computer system during the pre-boot phase of the first computer system, the debugger to operate from a firmware environment of the first computer system, the debugger executed by the first computer system independently of an operating system (OS) of the first computer system;  
initializing a communication channel of the first computer system to enable a second computer system to be communicatively coupled to the first computer system;  
communicatively coupling a second computer system to the first computer system at the communication channel;  
entering the debugger in response to a debug event; ~~and~~  
~~examining the first computer system with the debugger~~  
gathering machine state information about the first computer system with the debugger; and  
sending the machine state information to the second computer system from the first computer system.
2. Canceled.
3. (Currently Amended) The method of claim [[2]] 1 wherein the debug event comprises receiving a debug command from the second computer system at the first computer system.
4. (Original) The method of claim 3, further comprising setting a watchdog timer at the first computer system to periodically check for the debug command at the communication channel.

5. (Currently Amended) The method of claim [[2]] 3 wherein the debug event comprises receiving a wake-event from the second computer system at the first computer system.
6. (Original) The method of claim 5 wherein the wake-event comprises a Wake-on Local Area Network (LAN) packet.
7. (Original) The method of claim 5 wherein an operating system executing on the first computer system is hung.
8. (Original) The method of claim 1 wherein the debug event comprises detecting an exception that invokes an exception handler installed on the first computer system.
9. Canceled.
10. (Original) The method of claim 1, further comprising communicatively coupling a third computer system to the second computer system, the third computer system to interact with the debugger via the second computer system.
11. Canceled.
12. (Currently Amended) The method of claim 1 wherein the debugger is configured to operate ~~operable~~ during the pre-boot phase, the OS runtime phase, and an after-life phase of the first computer system.
13. (Currently Amended) ~~An article of manufacture comprising:  
a machine-accessible medium including a plurality of instructions which when executed perform operations comprising:~~  
An article of manufacture having a plurality of instructions stored thereon, the plurality of instructions to be executed independently of an operating system executed on the computer system, the instructions including:

initializing a debugger in a first computer system during the pre-boot phase of the first computer system, the debugger to operate from a firmware environment of the first computer system;

entering the debugger in response to a debug event from a second computer system communicatively coupled to the first computer system; and

~~examining the computer system with the debugger~~

gathering machine state information about the first computer system with the debugger; and

sending the machine state information to the second computer system from the first computer system.

14. (Original) The article of manufacture of claim 13 wherein the debug event comprises detecting an exception that invokes an exception handler installed on the computer system.

15. (Original) The article of manufacture of claim 13 wherein the debug event comprises receiving a wake-event from a second computer system communicatively coupled to the computer system.

16. Canceled.

17. Canceled.

18. Canceled.

19. (Original) The article of manufacture of claim 13 wherein the plurality of instructions to operate substantially in compliance with an Extensible Firmware Interface (EFI) specification.

20. (Currently Amended) A first computer system, comprising:

~~a first computer system, comprising:~~

a processor;  
a communication channel operatively coupled to the processor, the communication channel to be coupled to a second computer system; and  
at least one flash memory device operatively coupled to the processor, the at least one flash memory device including firmware instructions to be executed independently of an operating system (OS) executed on the first computer system, which when executed by the processor perform operations comprising:  
initializing a debugger in the first computer system during the pre-boot phase of the first computer system, the debugger to operate from a firmware environment of the first computer system;  
initializing the communication channel;  
entering the debugger in response to a debug event; ~~and~~  
~~examining the first computer system with the debugger~~  
gathering machine state information about the first computer system with the debugger; and  
sending the machine state information to the second computer system from the first computer system.

21. (Original) The system claim 20 wherein the debug event comprises detecting an exception that invokes an exception handler installed on the first computer system.
22. (Original) The system of claim 20 further comprising a second computer system communicatively coupled to the first computer system at the communication channel.
23. (Original) The system of claim 22 wherein the debug event comprises receiving a wake-event from the second computer system.
24. (Original) The system of claim 22 wherein the debug event comprises receiving a debug command from the second computer system.

25. (Original) The system of claim 22 further comprising a third computer system communicatively coupled to the second computer system, the third computer system to interact with the debugger via the second computer system.

26. Canceled.

27. (Currently Amended) The system of claim 20 wherein the debugger is configured to operate ~~operable~~ during the pre-boot phase, an operating system (OS) runtime phase, and an after-life phase of the first computer system.

28. (Original) The system of claim 20 wherein the firmware instructions to operate substantially in compliance with an Extensible Firmware Interface (EFI) specification.

29. (New) A method, comprising:

initializing a debugger in a first computer system during the pre-boot phase of the first computer system, the debugger to operate from a firmware environment of the first computer system, the debugger executed by the first computer system independently of an operating system of the first computer system;

initializing a communication channel of the first computer system to enable a second computer system to be communicatively coupled to the first computer system;

communicatively coupling a second computer system to the first computer system at the communication channel;

entering the debugger in response to a debug command from the second computer system at the first computer system;

setting a watchdog timer at the first computer system to periodically check for another debug command at the communication channel; and

examining the first computer system with the debugger.

30. (New) The method of claim 29 wherein the entering the debugger in response to the debug command includes entering the debugger during the pre-boot phase, an

operating system (OS) runtime phase, and an after-life phase of the first computer system.

31. (New) The method of claim 29 wherein an operating system executing on the first computer system is hung.